Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 155 (2019) 369–377

Procedia Computer Science

www.elsevier.com/locate/procedia

The 14th International Conference on Future Networks and Communications (FNC)
August 19-21, 2019, Halifax, Canada

Cloud Acknowledgment Scheme for a Node Network

Siddardha Kaja[a,*], Elhadi Shakshuki[a], Ansar Yasar[b]

[a]Jodrey School of Computer Science, Acadia Unversity, Wolfville, Nova Scotia, B4P2R6, Canada
[b]Transportation Research Institute, B-3500 Hasselt, Hasselt University, Belgium

Abstract

Recently, wireless devices are rapidly added to existing networks. This growth is due to abrupt development in technology and change of lifestyle. Due to the distribution nature of these networks, it is essential to mention that there is a substantial increase in the number of attacks as the network is expanding. By virtue of such trend, we are interested in bringing back centralization of a network even in wireless networks to deal with such attacks. In this paper, we propose a scheme called Cloud ACKnowledgement Scheme (CACKS) to strengthen a wireless network by fetching cloud as a monitoring tool. To validate our proposed approach, we performed several experiments using OMNET++ 5.4.1. The outcome of these experiments shows that the proposed scheme strongly monitors and protects the network from attackers while supporting unrestricted mobility.

Keywords: Cloud Acknowledegment scheme(CACKS); Cloud Computing; Mobile Adhoc NETwork(MANET)

1. Introduction

Over the past few years, cloud computing provides high availability, rapid elasticity and multitenancy. In cloud computing, compute, storage, network resources are abstracted from the underlying physical hardware and offered as a service. In this paper, we are interested in utilizing cloud services to propose a centralized monitoring and interacting scheme with nodes in a Mobile Ad hoc Network (MANET) [5]. MANET is a collection of mobile nodes

* Corresponding author. Tel.: 1-902-585-1524.
E-mail address: 152576k@acadiau.ca

10.1016/j.procs.2019.08.052

which can communicate wirelessly with both a transmitter as well as a receiver. There are many unique characteristics for a MANET, which makes network security a critical role in MANET. With no clear secure boundaries, compromised nodes, no central management, and the exitance scalability issue, makes it vulnerable to various types of attacks. Changing the position of a node makes it difficult to prevent malicious activities in a MANET. On the other hand, MANET considers every node in the network is healthy and cooperative. Attackers can easily compromise the MANET by direct and indirect attacks or by active or passive attacks [11]. Taking these scenarios into consideration, it is crucial to develop a centralized intrusion detection system specially for MANETs. The main purpose of this research work presented in this paper is to propose a new acknowledgement scheme in an existing MANET communication using the cloud.

In the following sub-section, we concentrate on discussing basic concepts as a background information required for understanding our proposed research work.

1.1. Overview of Basic Concepts

In this section, we present the basic concepts, existing approaches and related concepts. This includes an overview of existing intrusion detection systems (IDS), cloud computing and cloud-MANET model.

A. IDS in MANETs

As discussed before, Nodes in MANETs assume that all the nodes are cooperative with each other to transmit data. Therefore, the security issues in MANETs results in some typical and dangerous attacks on the network. This assumption makes the MANET susceptible to achieve significant impact on the network with few compromised nodes. Identifying the compromised nodes is a challenge that needs to be accomplished. An IDS is necessary to complete this challenge to achieve better security for the MANETs.

1) *Watchdog and Pathrater* [15]: The author of this work Marti and his co-authors proposed a scheme called watchdog. It has two main components called watchdog and pathrater. Watchdog acts as an IDS by copying packets to be forwarded into a buffer and observing the response of the adjacent node to these packets. If the watchdog snoops that next node fails to transmit the packet in a predefined amount of time, the failure counter increases. Also, when the node's failure counter crosses the predefined threshold, watchdog reports the nodes as malicious to the pathrater, then the pathrater cooperate with the routing protocols to avoid those malicious nodes.

2) *TWO ACKnowledgement scheme (TWOACK)* [16]: TWOACK is proposed by Liu and his colleagues. It is aimed to resolve receiver collision and limited transmitted power, which are some of the two of the major drawbacks of Watchdog scheme. Unlike Watchdog scheme, TWOACK detects the misbehaving links instead of misbehaving nodes by acknowledging every data packet transmitted over every three consecutive nodes. Every alternative node from source is supposed to send back an acknowledgement packet to a node that is two hops away from it down the route.

3) *Advanced ACKnowledgement scheme (AACK)* [17]: Sheltami with his colleagues proposed a new scheme that depend upon TWOACK called Advanced Acknowledgement scheme. It is also an acknowledgement-based scheme. It is a fusion of TACK (like TWOACK) and an end-to-end acknowledgement scheme called ACKnowledgement (ACK). In this scheme, when a sender sends a packet to the intermediate nodes, they must forward the packet until the destination node receives the data packet. When the destination node receives the packet, it sends an acknowledgement packet back to the sender in the reverse order using the same route. There is a predefined period in which the sender should receive the acknowledgement packet. Otherwise, the sender sends out a TACK packet by switching into TACK mode.

4) *Enhanced Adaptive ACKnowledgement (EAACK)* [1]: Shakshuki with his group developed Enhanced Adaptive Acknowledgement, which is an enhanced version of AACK. EAACK overcame the problem of corrupt and false acknowledgment by introducing digital signature into the scheme. A fixed length message digest is generated through a predetermined hash function for every message. The sender node transmits the message by applying its own private key on the message digest. When the destination node receives the message, it computes the message digest using a predetermined hash function. The authenticity of the message is verified by applying the public key of the sender node.

B. Cloud Computing

Cloud Computing is a popular computing paradigm. Cloud computing offers its services in three different models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS). In this research work we focus in using IaaS.

Cloud-MANET model is a combination of a MANET and a public cloud. The communication in a network occurs between nodes by discovering and connecting to the nearby nodes with no centralized infrastructure [7]. This happens with the help of Wi-Fi or cellular network [6]. In this proposed scheme, nodes use cloud services to find the nearby nodes and connect to them. Moreover, the scheme cooperates with routing policy to avoid malicious nodes.

2. Scheme Description

In this section, we describe our proposed Centralized ACKnowledgement Scheme (CACKS) in detail. The new approach described in this research paper is an ongoing research and based on our previous work described in [1] [12]. Where, the main pillar of CACKS is proposed and evaluated through simulation. In this paper, we enhance our previous contribution with the introduction of cloud to centralize the whole node network.

2.1 Assumptions

We assume that the MANET is in the range of cloud and cloud is neither malicious nor faulty. Since shortest paths avoid many problems and many cloud providers have been building out data distribution infrastructure and involving in direct interaction with clients. Moreover, many of the clients are single hop away (considering AS pathlengths) with development of recent techniques by google, Microsoft, and other cloud service providers [10]. We strongly believe that we can achieve better centralized MANET monitoring system by utilizing cloud services.

2.2 CACKS

In our proposed approach, we use two types of packets for communication in the cloud-MANET model. Firstly, a data packet which is a medium to carry data from one node to another node in a network. It is a datagram that consists of a header, payload and a footer. This is the information required by any node to process and forward the packet to its destination [1] [14].

Secondly, an acknowledgement packet, which is a conformation message packet that a receiver sends, to indicate that it received the data packet. Finally, Negative ACKnowledgement packet (NACK), it is sent to declare some malicious activity. NACK is usually sent to the sender over Real Time Control Protocol (RTCP). Later, the sender decides whether to resend the data packet again or not.

As previously stated, ACK is essentially an end-to-end acknowledgement scheme that we modified based on the following procedure. To start, the first sender node sends an acknowledgement packet to the cloud. A destination node's address is included in this acknowledgement packet's structure as shown in the Fig. 1.

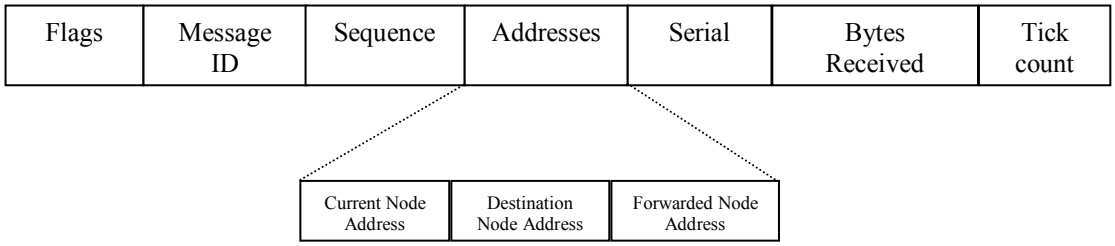

Fig. 1. Acknowledgement packet structure.

Attaching destination Node D address ensures that the cloud keeps track of the data packet from the sender Node S until it reaches the destination Node D. Next, the cloud sends an acknowledgement message to the sender Node S, signaling to proceed. sender Node S then forwards the data packet to the neighboring Node A, which then sends an acknowledgement packet to the cloud that this node has received the message. This acknowledgement packet from intermediate Node A includes the address of the neighboring intermediate node that it was forwarded to. This step assists the cloud in determining the location of the data packet. This process repeats until the data packet reaches the destination Node D. When the destination Node D receives the data packet, it sends an acknowledgement packet to the cloud that it received the packet. The cloud then notifies the sender Node S that the data packet reached its destination, as shown in Fig. 2.

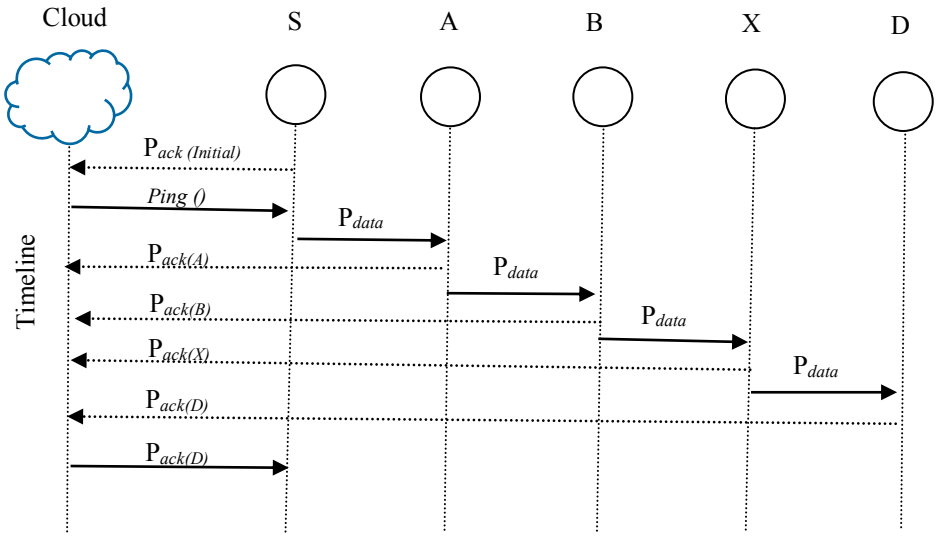

Fig. 2. System control: The system flow of how CACKS scheme works.

The role of cloud in a single data packet transmission is described in Algorithm 1.

Algorithm 1

1. Cloud receives $P_{ack(initial)}$ from Node S
   1.1 Parse the $P_{ack(initial)}$
   1.2 Validate the $P_{ack(initial)}$
      1.2.1 IF (valid) Then
      1.2.2 Store in the buffer
      1.2.3 IF (invalid) Then
      1.2.4 Send NACK to Node S
   1.3 Process the $P_{ack(initial)}$
2. Ping Node S
3. Count = 0, MAX = Predefined Time (PT)
4. Loop Begins: Count = Count + 1
   4.1 IF (Count ≤ PT) Then
   4.2 Wait;
   4.3 IF (Count is Greater than PT)
   4.4 Send NACK to Node S
   4.5 Receives $P_{ack(A)}$ from Node A
   4.6 Parse the $P_{ack(A)}$
   4.7 Validate the $P_{ack(A)}$
      4.7.1 IF (valid) Then
      4.7.2 Store in the buffer
      4.7.3 IF (invalid) Then
      4.7.4 Send NACK to Node S and Node A
   4.8 Process the $P_{ack(A)}$
   4.9 Store in the buffer
5. Receives $P_{ack(D)}$ from Node D
   5.1 Parse the $P_{ack(D)}$
   5.2 Validate the $P_{ack(D)}$
      5.2.1 IF (valid) Then
      5.2.2 Store in the buffer
      5.2.3 IF (invalid) Then
      5.2.4 Send NACK to Node S and Node D
   5.3 Process the $P_{ack(D)}$
   5.4 Forward ACK from Node D to Node S

The sender's role in a single data packet transmission is described in Algorithm 2.

Algorithm 2

1. Node A Creating $P_{ack(initial)}$
   1.1 Include Node S address to $P_{ack(initial)}$
   1.2 Include Node D address to $P_{ack(initial)}$
   1.3 Include Node A address to $P_{ack(initial)}$
2. Node S sends $P_{ack(initial)}$ to the Cloud
3. Loop begins Count = 0; MAX = Predefined Time (PT)
4. Count = Count + 1;
   4.1 If (Count < PT & Node A receives Ping () from Cloud) Then
   4.2 Proceed
   4.4 If (Count >= PT & Node A doesn't receive Ping() from Cloud) Then

4.4 Send P_{ack}(initial) again
5. Node S creates P_{data}
   5.1 Include Node S address to P_{data}
   5.2 Include Node D address to P_{data}
   5.3 Include Node A address to P_{data}
6. Node S Forwards P_{data} to Node A
7. If (Receives NACK) Then
   7.1 Take another route
   7.2 Resend P_{data} to another available node

Algorithm 3 reveals the intermediate node's responsibility in a single data packet transmission.

Algorithm 3

1. Node A Receives P_{data} from Node S
   1.1 Parse P_{data}
   1.2 Read Destination address

2. If(Node A can Reach Node B) Then
   2.1 Creating $P_{ack(A)}$
      2.1.1 Include Node A address to $P_{ack(A)}$
      2.1.2 Include Node D address to $P_{ack(A)}$
      2.1.3 Include Node B address to $P_{ack(A)}$
3. Node A sends $P_{ack(A)}$ to the Cloud
4. If (Node A cannot reach Node B) Then
   4.1 Sends NACK to the Cloud
   4.2 Create new route
   4.3 Sends P_{data} to new intermediate Node X
   4.4 Creating $P_{ack:new(A)}$
      4.1.1 Include Node A address to $P_{ack:new(A)}$
      4.1.2 Include Node X address to $P_{ack:new(A)}$
      4.1.3 Include Node D address to $P_{ack:new(A)}$
5. Node A sends $P_{ack:new(A)}$ to the Cloud

Centralized acknowledgement scheme secures a MANET by detecting less secured nodes in the following ways. All the acknowledgement packets are collected and stored in the cloud. If the cloud doesn't receive another acknowledgement packet within a predefined period from receiving the data packet from one intermediate node, cloud considers that the next intermediate node is faulty and accordingly sends a NACK to the sender. As shown in Fig. 3.

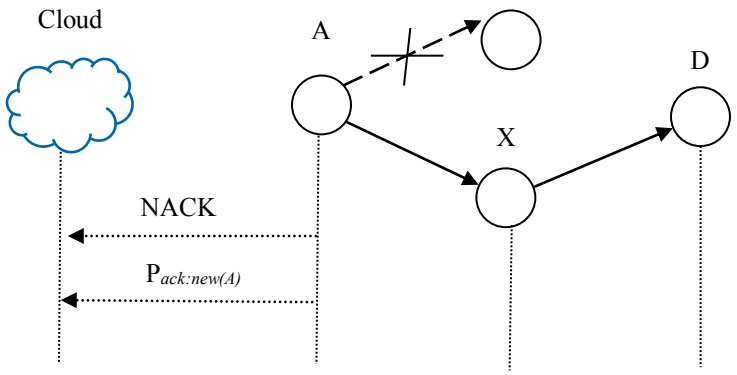

Fig. 3. Failed packet transmission: Node A reroutes the data packet to node X if it fails to connect with node B.

If a Node A couldn't forward the data packet to the next intermediate Node B, Node A sends a NACK to the cloud and informing the cloud that the next neighboring Node B is not reachable. Then, Node A will decide the new route for data packet to reach its destination Node D. These acknowledgement packets collected in the cloud are used to determine the failure counter of a certain node. If the failure counter exceeds the pre-determined limit, then the node is considered malicious. Moreover, the cloud cooperates with the routing mechanism to avoid these failure nodes. False misbehavior reporting node problem is solved since only the cloud can report the misbehaving nodes.

3. Scheme Description

In this section, we describe our simulation environment and methodology as well as our achieved results. We show a concrete example of CACKS application and its results.

3.1 Simulation Methodologies

Since our proposed scheme is a new alternate approach for existing IDS. We simulate our CACKS scheme in a simulator where the cloud gathers all the acknowledgement packets from the nodes. Whereas, the proposed nodes can concentrate on forwarding the data packets. We demonstrate two scenario networks to simulate different types of attacks.

In the first scenario, we simulated a basic packet dropping instance. A failure node simply drops all the packets as it receives them. In the second scenario, we simulate a situation where a node cannot connect with the neighboring node. The purpose of the latter scenario is to demonstrate the rerouting ability of the node when the things go south.

3.2 Simulation Configurations

Our simulation is conducted within the OMNET++ 5.4.1 environment with INET framework in a windows 10 operating system [19]. The system is running on a laptop with Intel(R) Core (TM) i7-8750H CPU @ 2.20GHz and 16-GB RAM.

To demonstrate our simulation in a better approach, we implemented the default scenario settings in OMNET++ 5.4.1. The default configuration identifies 4 nodes in a flat space with a size of 800m × 800m physical layer as well as 802.11 MAC layer are included in the INET framework of OMNET++.

User Datagram Protocol (UDP) traffic with constant bit rate is maintained with a packet size of 1 MB. For each scenario, we ran the network ten times and calculate the average performance.

To measure the performance of CACKS, we use a metric called Packet Delivery Ratio (PDR) which defines the ratio of number of packets received by the destination node to the number of packets sent by the source node.

3.3 Simulation Results

This section presents the performance results of our proposed scheme. CACKS is designed to centralize the network, there by cloud identifying the malicious node is very important feature. As shown in Fig.3, cloud identifies the malicious node while sender Node S attempts to a transmit a data packet in a network called CACK where the Node D is the destination. All the devices are in the same network and in the range of cloud.

Each time we ran the CACKS network, the cloud responds to the sender by helping the data packet to reach its destination. We place a malicious node in our network which is not able to communicate with any surrounding node When Node A cannot reach Node B, the cloud successfully identified the details of the malicious node. When we ran the scenario where the Node B is not reachable, Node A is successfully rerouting the Pdata to Node X every time. PDR is 100 percent in this demonstration of our proposed scheme every time.

As intimated earlier, we setup a flat space in the CACKS network. a radio medium is stationed for the communication of wireless devices. An IPv6 network configurator is used as this protocol supports growth in the number of devices and the raise in data traffic in a network. Cloud is placed in a position of equal distance to all the nodes in this network so that better single hop communication is possible. Node S is the sender node while Node D is the destination node and Node A, Node B, Node X are the intermediate nodes as represented in Fig .4.

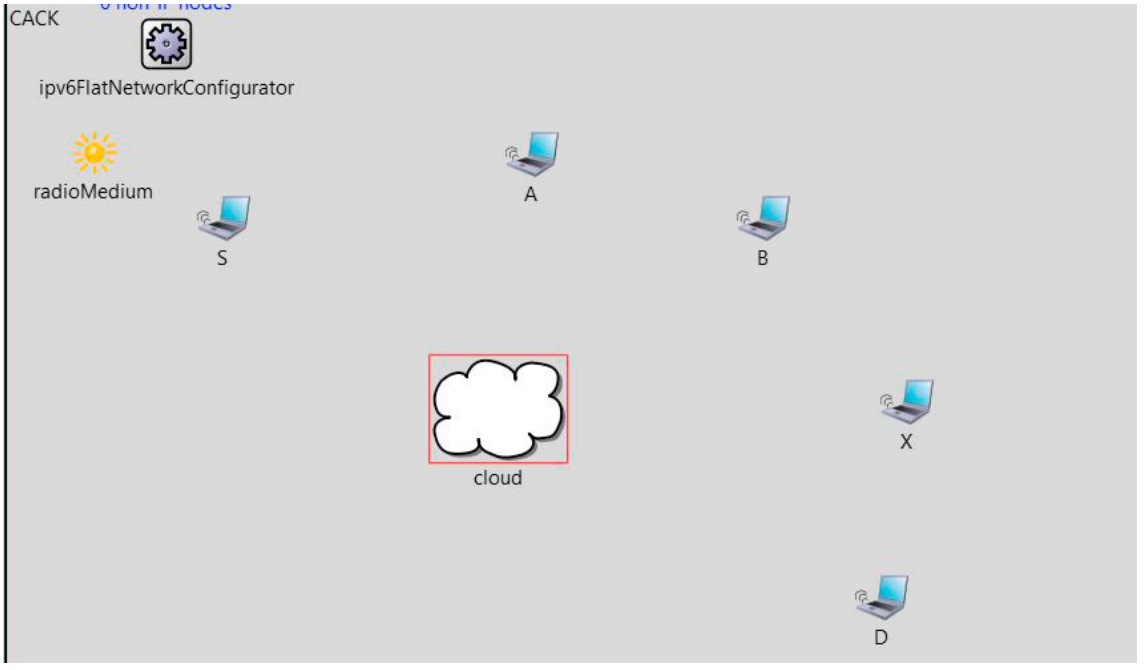

Fig. 4. Comprehensive snapshot of CACKS network in OMNET++ IDE.

4. Conclusion and Future Work

Centralizing the network is normally a desired approach to provide more benefit to the security of a network. In our proposed approach, we proposed a centralized acknowledgement scheme called CACKS. It was demonstrated as to how the cloud plays a crucial role as a monitoring system to achieve the centralized network by collecting acknowledgement packets and detecting the malicious nodes.

In furtherance of our research work, we plan to test the possibility of CACKS running to centralize different types of networks like VANETs and IoT where a centralized monitoring system is substantial. We also plan to enhance the scope of CACKS by modifying the technique for multi-hop networks.

References

[1] Elhadi M. Shakshuki, Nan Kang and Lester Tarek R. Sheltami. (2013) "EAACK—A Secure Intrusion-Detection System for MANETs*" IEEE Transaction and Industrial Electronics, Vol. 60, No. 3.*

[2] Tarek Sheltami, Abdulsalam Basabaa, Elhadi M. Shakshuki. "A3ACKs: adaptive three acknowledgments intrusion detection system for MANETs" *Journal of Ambient Intelligence and Humanized Computing August 2014, Volume 5, Issue 4, pp 611–620.*

[3] Babatunji Omoniwa, Riaz Hussain, Muhammad Awais Javed, Safdar H. Bouk, Senior Member, IEEE and Shahzad A. Malik. (2018) "Fog/Edge Computing-based IoT (FECIoT): Architecture, Applications, and Research Issues." *IEEE Internet of Things Journal*.

[4] Ihsan Ali, Abdullah Gani, Ismail Ahmedy, Ibrar Yaqoob, Suleman Khan, and Mohammad Hossein Anisi. (2018) "Data Collection in Smart Communities Using Sensor Cloud: Recent Advances, Taxonomy, and Future Research Directions." *IEEE Communications Magazine*

[5] Mohammad Aazam, Eui-Nam Huh, Marc St-Hilaire, Chung-Horng Lung and Ioannis Lambadaris (2015) "Cloud of Things: Integration of IoT with Cloud Computing"'.

[6] Alshareef, H. N., & Grigoras, D. (2014). "Mobile Ad-hoc Network Management in the Cloud*." IEEE 13th International Symposium on Parallel and Distributed Computing.*

[7] Tanweer Alam, Mohamed Benaida. (2018) "The Role of Cloud-MANET Framework in the Internet of Things (IoT)" *iJOE – Vol. 14, No. 12.*

[8] Pablo Pu˜nal Pereira, Jens Eliasson, Rumen Kyusakov, Jerker Delsing and Asma Raayatinezhad. (2018) "EnablingCloud-connectivityforMobileInternetofThingsApplications" *2013 IEEE Seventh International Symposium on Service-Oriented System Engineering.*

[9] Sebastian Dombrowski, Tatiana Ermakova, Benjamin Fabian "Graph-Based Analysis of Cloud Connectivity at the Internet Protocol Level"

[10] Yi-Ching Chiu, Brandon Schlinker, Abhishek Balaji Radhakrishnan, Ethan Katz-Bassett, Ramesh Govindan. (2015) "Are We One Hop Away from a Better Internet?" *IMC'15, October 28–30, 2015, Tokyo, Japan.*

[11] Kiriti Gupta, Dr. Pradeep K. Mittal (2017) "An Overview of Security in MANET" *International Journals of Advanced Research in Computer Science and Software Engineering ISSN: 2277-128X (Volume-7, Issue-6).*

[12] Elhadi M. Shakshuki, Nan Kang and Lester Tarek R. Sheltami. (2010) "Detecting Misbehaving Nodes in MANETS" in *Proc. 12th Int. Conf. iiWAS, Paris, France, Nov. 8-10, 2010, pp. 216-222.*

[13] Chirag Modi, Dhiren Patel, Bhavesh Borisaniya, Hiren Patel, Avi Patel, Muttu Krishnan Rajarajan. (2012) "A survey of Intrusion detection techniques in cloud" *Journal of Network and Computer Applications 36 (2013) 42–57.*

[14] David E. Halasz (2001) "Packet Assembly" *United States Patent, Patent No.: US 7,039,068 B1.*

[15] S. Marti, T. J. Giuli, K. Lai, and M. Baker. (2000) "Mitigating routing misbehavior in mobile ad hoc networks" *in Proc. 6th Annu. Int. Conf. Mobile compute. Netw., Boston, MA,*2000, pp. 255-265.

[16] K. Liu, Jing Deng, Pramod K. Varshney and Kashyap Balakrishnan. (2007) "An Acknowledgment-Based Approach for the Detection of Routing Misbehavior in MANETs" *IEEE Transactions on Mobile Computing, Vol. 6, no. 5, May 2007.*

[17] T. Sheltami, A. Al-Roubaiey, Elhadi M Shakshuki and A Mahmoud. (2009) "Video transmission enhancement in presence of misbehaving nodes in MANETs" *Int. J. Multimedia Syst., vol. 15, no. 5, pp. 273-282, Oct. 2009.*

[18] D. Snowdon, Natalie S. Glance. (2003) "Decentralized Network System" *United States Patent, Patent No.: US 6,671,737 B1.*

[19] Omnet++, Omnest, UAE. https://omnetpp.org Accessed: Jan 26th, 2019.